UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE CIENCIAS FÍSICAS

DEPARTAMENTO DE FÍSICA DE LA TIERRA Y ASTROFÍSICA

TRABAJO DE FIN DE GRADO

Código de TFG: FTA16

Elaboración de un modelo numérico de interior estelar

Development of a stellar-interior numerical model

Supervisor/es: Sergio Pascual Ramírez, Nicolás Cardiel López

Nuno Cerviño Luridiana

Grado en Física

Curso académico 2024-25

Convocatoria ordinaria

Calificación: 9.9 (Matricula de Honor)

Resumen:

El desarrollo de modelos numéricos del interior estelar es una herramienta fundamental en astrofísica, ya que permite estudiar fenómenos de otro modo inaccesibles como las distribuciones de magnitudes físicas en función del radio, el transporte de energía o la ubicación y extensión de las zonas convectivas y radiativas. En este trabajo se construye un modelo teórico simplificado de una estrella de masa intermedia en la secuencia principal, incorporando los principales mecanismos físicos que rigen su estructura. A partir de este marco, se desarrolla un código computacional que resuelve las ecuaciones estructurales y permite obtener perfiles internos de presión, temperatura, luminosidad, masa, densidad, rimo de generación de energía y opacidad en función del radio estelar. Además, se explora la sensibilidad del modelo a distintos parámetros, como la metalicidad, y se analizan los resultados obtenidos en relación con predicciones teóricas conocidas, evaluando su potencial como base para modelos estelares más avanzados.

Abstract:

The development of numerical models of stellar interiors is a fundamental tool in astrophysics, as it allows the study of otherwise inaccessible phenomena such as the distribution of physical quantities as a function of radius, energy transport, and the location and extent of convective and radiative zones. In this thesis, a simplified theoretical model is constructed for an intermediate-mass star on the main sequence, incorporating the main physical mechanisms that govern its structure. Based on this framework, a computational code is developed to solve the structural equations and obtain internal profiles of pressure, temperature, luminosity, mass, density, energy generation rate, and opacity as functions of the stellar radius. Additionally, the model's sensitivity to various parameters, such as metallicity, is explored, and the results are analyzed in relation to known theoretical predictions, judging its potential as a foundation for more advanced stellar models.

Contents

1 Introduction

1.1 Starting point and objective

Numerical modeling of stellar interiors requires formulating the fundamental system of differential equations that governs them. As a first step, stars are assumed to be isolated, static, and spherically symmetric, as well as to exhibit adiabatic convection, local thermodynamic equilibrium, and the presence of N different chemical species. Under these assumptions, the system to be solved consists of $4 + N$ partial differential equations. In this context, the system developed in Kippenhahn et al. (2012) [1] is presented here, without a detailed discussion of its components since, as explained in Section 1.2, most of these equations will not be necessary.

$$\frac{\partial m}{\partial r} = 4\pi r^2 \rho \qquad \frac{\partial T}{\partial r} = \begin{cases} -\dfrac{3}{16\pi ac}\dfrac{\kappa\rho\ell}{r^2 T^3} & \text{if} \quad \nabla_{\text{rad}} \leq \nabla_{\text{ad}} \\ -\dfrac{Gm\rho}{r^2}\dfrac{T}{P}(\nabla_{\text{ad}} + \Delta\nabla) & \text{if} \quad \nabla_{\text{rad}} > \nabla_{\text{ad}} \end{cases}$$

$$\frac{\partial P}{\partial r} = -\frac{Gm}{r^2}\rho - \rho\frac{\partial^2 r}{\partial t^2}$$

$$\frac{\partial \ell}{\partial r} = 4\pi r^2 \rho(\varepsilon - \varepsilon_\nu - T\frac{\partial s}{\partial t}) \qquad \frac{\partial X_i}{\partial t} = \frac{A_i m_u}{\rho}\left[-\sum_j (1+\delta_{ij}) r_{ij} + \sum_{k,l} r_{kl,i}\right]$$

From this basis, the objective is to develop a practical, theory-based framework to model a star from its total mass M_{total} and chemical composition $\{X_i\}$, expressed in terms of the hydrogen (X) and helium (Y) mass fractions. To reduce the complexity of the problem, the model will focus on intermediate-mass stars with a convective core and a radiative envelope. It will also allow for adjustment of parameters such as the total radius R_{total}, total luminosity L_{total}, and central temperature T_c. The outcome should enable the characterization of the radial profiles of pressure P, temperature T, luminosity ℓ, mass m, density ρ, energy generation rate ε, and opacity κ as functions of the stellar radius r.

1.2 Basic assumptions

To develop a numerical model of stellar interiors, it is necessary to introduce a set of basic assumptions that significantly simplify the integration of the differential equation system.

- Time-dependent terms can be neglected, as the stars that are being considered are both in hydrostatic and thermal equilibrium and their chemical composition changes over long timescales.

- The chemical composition is assumed to be constant throughout the star, and the superadiabaticity $\Delta\nabla$ is taken to be zero in the stellar interior.

- The stellar material behaves as a fully ionized, non-degenerate ideal gas. This implies treating the stellar interior as a monoatomic gas ($\gamma_{\text{ad}} = 5/3$), with an adiabatic temperature gradient given by $\nabla_{\text{ad}} = 1 - 1/\gamma_{\text{ad}} = 0.4$. Defining metallicity as the mass fraction of heavy elements, $Z = 1 - X - Y$, a simple expression for the mean molecular weight can be derived.

$$\mu = \frac{1}{2X + 0.75\,Y + 0.5Z} \tag{1}$$

Furthermore, denoting Avogadro's number as $N_A = 6.02214 \times 10^{23}\,\mathrm{g}^{-1}$ and Boltzmann's constant as $k = 1.38065 \times 10^{-16}\,\mathrm{erg\,K}^{-1}$, the ideal gas equation of state can be applied:

$$\rho = \frac{\mu}{N_A k}\frac{P}{T} \tag{2}$$

- In the inner layers of the star, where convection takes place, the stellar material can be approximated as a polytropic gas with index $n = 1.5$, which corresponds to a fully convective region. Accordingly, the following polytropic relation applies in these zones:

$$P = K'\,T^{\gamma/(\gamma-1)} = K'\,T^{2.5} \tag{3}$$

Here, the value of K' is estimated using the magnitude of P and T in the innermost radiative layer of the star and its assumed as a constant.

- Opacity in the star is modeled using Kramers' law for bound-free electronic transitions, the main opacity source in intermediate-mass stars, as noted by Schwarzschild (1965, p. 70) [2].

$$\kappa = 4.34 \times 10^{25} g_{\mathrm{bf}} Z(1+X)\frac{\rho}{T^{3.5}} \tag{4}$$

In this expression, g_{bf} is the Gaunt factor for bound-free transitions, which, according to Novotny (1973, p. 449) [3], takes the value $g_{\mathrm{bf}} = 1/3.162$.

- The energy generation rate can be approximated by a power-law expression:

$$\varepsilon = \varepsilon_1 X_1 X_2 \rho \left(\frac{T}{10^6\,\mathrm{K}}\right)^{\nu} \tag{5}$$

where ε_1, ν, and the mass fractions X_1 and X_2 depend on the dominant nuclear reaction chain (either the proton-proton chain or the CN cycle) as well as on the temperature. The values of ε_1 and ν are listed in Table 1, while the mass fractions satisfy $X_1 = X_2 = X$ for the proton-proton chain, and $X_1 = X$, $X_2 = \frac{1}{3}Z$ for the CN cycle.

For a given temperature, the energy generation rate is computed for both reaction chains, and the value corresponding to the more efficient process is adopted. If the temperature lies outside the operational range of both cycles, energy generation is assumed to be negligible.

ε_{pp}			ε_{CN}		
$T/10^7$	$\log_{10}\varepsilon_1$	ν	$T/10^7$	$\log_{10}\varepsilon_1$	ν
$0.40-0.60$	-6.84	6.0	$1.20-1.60$	-22.2	20
$0.60-0.95$	-6.04	5.0	$1.60-2.25$	-19.8	18
$0.95-1.20$	-5.56	4.5	$2.25-2.75$	-17.1	16
$1.20-1.65$	-5.02	4.0	$2.75-3.60$	-15.6	15
$1.65-2.40$	-4.40	3.5	$3.60-5.00$	-12.5	13

Table 1: Parameters for calculating the energy generation rate for both the proton-proton chain and the CN cycle (quantities expressed in CGS). Table taken from Schwarzschild (1965, p. 83) [2].

2 Methodology

2.1 Fundamental equations

Applying the hypotheses described in Section 1 to the original system of partial differential equations leads to a significant reduction in the complexity of the problem. The key equations used to build the model are summarized in Table 2.

Radiative transport	Convective transport
$\frac{\partial m}{\partial r} = 4\pi \frac{\mu}{N_A k} \frac{P}{T} r^2$	$\frac{\partial m}{\partial r} = 4\pi \frac{\mu}{N_A k} K' T^{1.5} r^2$
$\frac{\partial P}{\partial r} = -\frac{G}{N_A k} \mu \frac{P}{T} \frac{m}{r^2}$	$\frac{\partial P}{\partial r} = -\frac{G}{N_A k} \mu K' T^{1.5} \frac{m}{r^2}$
$\frac{\partial \ell}{\partial r} = 4\pi (\frac{1}{N_A k})^2 \varepsilon_1 X_1 X_2 \mu^2 \frac{P^2}{T^2} (\frac{T}{10^6})^\nu r^2$	$\frac{\partial \ell}{\partial r} = 4\pi (\frac{1}{N_A k})^2 \varepsilon_1 X_1 X_2 \mu^2 K'^2 T^3 (\frac{T}{10^6})^\nu r^2$
$\frac{\partial T}{\partial r} = -\frac{3}{16\pi ac} 4.34 \times 10^{25} g_{\text{bf}} (\frac{1}{N_A k})^2 Z(1+X) \mu^2 \frac{P}{T^{8.5}} \frac{\ell}{r^2}$	$\frac{\partial T}{\partial r} = -0.4 \frac{G}{N_A k} \mu \frac{m}{r^2}$

Table 2: Fundamental equations for stellar interiors after applying the assumptions from Section 1.

2.2 Units

Since the numerical integration of the stellar model involves multiple iterations and calculations, it is convenient to adopt a unit system in which most quantities are close to unity. The chosen units are presented in Table 3 and will be referred to as the model unit system from this point forward. Applying this unit system to the equations in Table 2 yields the set of formulas that will be implemented in the model's code.

2.2.1 Equations for radiative transport

$$\frac{\partial m}{\partial r} = C_m \frac{P}{T} r^2 \qquad C_m = 0.01523\mu \tag{6}$$

$$\frac{\partial P}{\partial r} = -C_p \frac{P}{T} \frac{m}{r^2} \qquad C_p = 8.084\mu \tag{7}$$

$$\frac{\partial \ell}{\partial r} = C_\ell P^2 T^{\nu-2} r^2 \qquad C_\ell = 0.01845 \varepsilon_1 X_1 X_2 10^\nu \mu^2 \tag{8}$$

$$\frac{\partial T}{\partial r} = -C_t \frac{P^2}{T^{8.5}} \frac{\ell}{r^2} \qquad C_t = 0.01679 Z(1+X)\mu^2 \tag{9}$$

2.2.2 Equations for convective transport

$$\frac{\partial m}{\partial r} = C'_m K' T^{1.5} r^2 \qquad C'_m = 0.01523\mu \tag{10}$$

$$\frac{\partial P}{\partial r} = -C'_p K' T^{1.5} \frac{m}{r^2} \qquad C'_p = 8.084\mu \tag{11}$$

$$\frac{\partial \ell}{\partial r} = C'_\ell K'^2 T^{3+\nu} r^2 \qquad C'_\ell = 0.01845 \varepsilon_1 X_1 X_2 10^\nu \mu^2 \tag{12}$$

$$\frac{\partial T}{\partial r} = -C'_t \frac{m}{r^2} \qquad C'_t = 3.234\mu \tag{13}$$

Quantity	Symbol	Units
Radius	r	10^{10} cm
Pressure	P	10^{15} dyn cm^{-2}
Temperature	T	10^{7} K
Mass	m	10^{33} g
Luminosity	ℓ	10^{33} erg s^{-1}
Density	ρ	g cm^{-3}
Energy generation rate	ε	erg g^{-1} s^{-1}
Opacity	κ	cm^2 g^{-1}

Table 3: Unit system adopted for the construction of the model.

2.3 Integration and computational development

To solve the proposed system, a hybrid method will be used, combining inward integration from the stellar surface and outward integration from the center, connecting both solutions at the boundary between the convective core and the radiative envelope. Additionally, adjusting the parameters R_{total}, L_{total}, and T_{c} ensures a smooth transition between these two regions.

2.3.1 Predictor–corrector algorithm

The numerical integration of the equations presented in Section 2.2 is accomplished using the method described by Novotny (1973, Chapter 8) [3]. In this approach, the star is divided into concentric spherical layers, with v_i and z_i defined as the dependent and independent variables, respectively, evaluated at layer i. The integration step is given by $h = z_{i+1} - z_i$, and $f_i = \left(\frac{\partial v}{\partial z}\right)_i$ represents the variation of v with respect to z at layer i. Using these elements, a power series expansion is employed to predict the value of the variable v in successive layers.

$$v_{i+1} = v_i + h\left(\frac{\partial v}{\partial z}\right)_i + \frac{1}{2}h^2\left(\frac{\partial^2 v}{\partial z^2}\right)_i + \frac{1}{6}h^3\left(\frac{\partial^3 v}{\partial z^3}\right)_i + \cdots \approx v_i + h\left(\frac{\partial v}{\partial z}\right)_i = v_i + hf_i \tag{14}$$

However, truncating the expansion at first order reduces accuracy. Since computing higher-order derivatives would greatly increase complexity, first-order finite differences are used instead.

$$\begin{aligned} {}^1\Delta_i &\equiv h(f_i - f_{i-1}) \\ {}^1\Delta_{i+1} &\equiv h(f_{i+1} - f_i) \\ {}^2\Delta_i \equiv {}^1\Delta_i - {}^1\Delta_{i-1} &= h(f_i - 2\,f_{i-1} + f_{i-2}) \\ {}^2\Delta_{i+1} \equiv {}^1\Delta_{i+1} - {}^1\Delta_i &= h(f_{i+1} - 2\,f_i + f_{i-1}) \end{aligned}$$

This leads to two expressions:

$$v_{i+1} = v_i + hf_i + a_1\,{}^1\Delta_i + a_2\,{}^2\Delta_i + a_3\,{}^3\Delta_i + \cdots \tag{15}$$

$$v_{i+1} = v_i + hf_{i+1} + b_1\,{}^1\Delta_{i+1} + b_2\,{}^2\Delta_{i+1} + b_3\,{}^3\Delta_{i+1} + \cdots \tag{16}$$

where the coefficients a_1, a_2, ... and b_1, b_2, ... are determined by ensuring that hf is a polynomial of the same degree as the highest order of the finite differences to be used. The number of retained differences should be kept moderate to avoid significant rounding errors and numerical instability.

Rather than using an excessive number of differences, it is preferable to decrease the interval size. This approach allows the derivation of both an open integration formula v_{i+1}^{est} and a closed one v_{i+1}^{cal}, where second-order differences are reserved for pressure estimation and luminosity calculations, while first-order differences suffice for the remaining variables.

$$v_{i+1} = v_i + hf_i + \frac{1}{2}\,{}^1\Delta_i + \frac{5}{12}\,{}^2\Delta_i \equiv v_{i+1}^{\text{est}} \tag{17}$$

$$v_{i+1} = v_i + hf_i - \frac{1}{2}\,{}^1\Delta_{i+1} - \frac{1}{12}\,{}^2\Delta_{i+1} \equiv v_{i+1}^{\text{cal}} \tag{18}$$

As formulated, the closed formula exhibits higher convergence than the open one. However, it requires an initial estimate of the variable being calculated. Thus, an iterative predictor-corrector method is used: a preliminary value v_{i+1}^{est} is estimated using equation (17), and then used to compute v_{i+1}^{cal} via equation (18). The loop continues while the relative error between the two values remains below a tolerance threshold $T = 0.0001$, updating v_{i+1}^{est} with the value of v_{i+1}^{cal} at each iteration.

$$\frac{|v_{i+1}^{\text{cal}} - v_{i+1}^{\text{est}}|}{v_{i+1}^{\text{cal}}} < T \tag{19}$$

It is important to note that implementing this algorithm requires the value of a variable and its derivatives at layer i and the two previous layers. Therefore, an initial method must be defined to estimate these values when starting integration both from the surface and from the center. Section 2.4 provides the approximate expressions used to compute the first few layers before starting the integration.

2.3.2 Integration from the surface

To avoid convergence issues in the outermost layers, the integration from the surface starts at $R_{\text{initial}} = 0.9\,R_{\text{total}}$. From this radius, the star is divided into 100 spherical layers, with a negative step size $h < 0$ due to the inward integration. Initially, the first three layers are computed assuming $m = M_{\text{total}}$, $\ell = L_{\text{total}}$, and using equations (22) and (23) from Section 2.4.1. Then, the derivatives are computed iteratively using equations (6), (7), (8), and (9), while star variables are determined using the predictor-corrector algorithm until the assumption of radiative transport becomes invalid. To identify this point, Novotny (1973, Chapter 6, Section 4-4) [3] introduces the parameter $n + 1$ and states that radiative transport holds for values *less than or equal to* 2.5. When this threshold is exceeded, integration from the center begins.

$$n + 1 = \frac{T_{i+1}^{\text{cal}}}{P_{i+1}^{\text{cal}}} \frac{(\frac{\partial P}{\partial r})_{i+1}}{(\frac{\partial T}{\partial r})_{i+1}} \leq 2.5 \tag{20}$$

2.3.3 Integration from the center

In contrast to the previous case, a positive step size $h > 0$ is defined as the integration proceeds outward. The three innermost layers are calculated using equations (24), (25), (26), and (27) from Section 2.4.2. Then, equations (10), (11), (12), and (13) are iterated along with the predictor-corrector algorithm until the last layer calculated from the surface is reached.

2.3.4 Outermost layers

Once the integration from the center is complete, the outermost layers are computed from R_{initial} to R_{total} using again equations (22) and (23) from Section 2.4.1.

2.3.5 Total relative error

As previously discussed, for a fixed total mass M_{total} and chemical composition $\{X_i\}$, different sets of initial values for the total radius R_{total}, total luminosity L_{total}, and central temperature T_{c} can lead to smoother or sharper transitions between the radiative and convective zones. Therefore, once the calculation is complete, it is necessary to evaluate the quality of the solution by comparing the results from both integrations at a common layer. For this purpose, the total relative error E is defined as a parameter to be minimized:

$$E = \sqrt{\left(\frac{P_{\text{rad}} - P_{\text{conv}}}{P_{\text{rad}}}\right)^2 + \left(\frac{T_{\text{rad}} - T_{\text{conv}}}{T_{\text{rad}}}\right)^2 + \left(\frac{m_{\text{rad}} - m_{\text{conv}}}{m_{\text{rad}}}\right)^2 + \left(\frac{\ell_{\text{rad}} - \ell_{\text{conv}}}{\ell_{\text{rad}}}\right)^2} \tag{21}$$

where the subscript "rad" indicates that the quantity was calculated in the integration from the surface, and "conv" means it was obtained from the integration starting at the center.

2.4 Initial values

Section 2.3 outlined the need to estimate tentative initial values both at the surface and in the innermost layers of the model star. The theoretical derivation of the results presented here can be found in Cardiel & Pascual (2024, Section 2.4) [4].

2.4.1 Initial values at the surface

Near the stellar surface, there is no energy production and the luminosity remains constant. Moreover, the density in these outer layers is significantly lower than in deeper regions, making the contribution of this zone to the total mass negligible. As a result, the conditions $m = M_{\text{total}}$ and $\ell = L_{\text{total}}$ hold with a high degree of accuracy. These assumptions allow the values of T and P to be determined along with m and ℓ. Expressed in the model's unit system, the following relations are obtained:

$$T = A_1 \left(\frac{1}{r} - \frac{1}{R_{\text{total}}}\right) \qquad A_1 = 1.9022 \mu M_{\text{total}} \tag{22}$$

$$P = A_2\, T^{4.25} \qquad A_2 = 10.645 \sqrt{\frac{1}{\mu Z(1+X)} \frac{M_{\text{total}}}{L_{\text{total}}}} \tag{23}$$

2.4.2 Initial values at the center

Given the central temperature T_{c}, it is assumed that a convective core is present. A power series expansion of the equations in Table 2 yields expressions that are valid in the innermost layers. Rewriting these equations using the model's units leads to:

$$m = B_1 K' T_{\text{c}}^{1.5} r^3 \qquad B_1 = 0.005077 \mu \tag{24}$$

$$\ell = B_2 K'^2 T_{\text{c}}^{3+\nu} r^2 \qquad B_2 = 0.006150 \varepsilon_1 X_1 X_2 10^{\nu} \mu^2 \tag{25}$$

$$T = T_{\text{c}} - B_3 K' T_{\text{c}}^{1.5} r^2 \qquad B_3 = 0.008207 \mu^2 \tag{26}$$

$$P = K' T^{2.5} \tag{27}$$

2.5 Algorithm flowchart

The procedures described throughout Section 2.3 are presented here in the form of a flowchart, indicating the computed quantities and the equations used at each step. Additionally, each successive step uses values calculated in previous layers. Although not explicitly shown in the diagram, equations (2), (4), and (5) are used at every layer to determine ρ, κ, and ε, respectively.

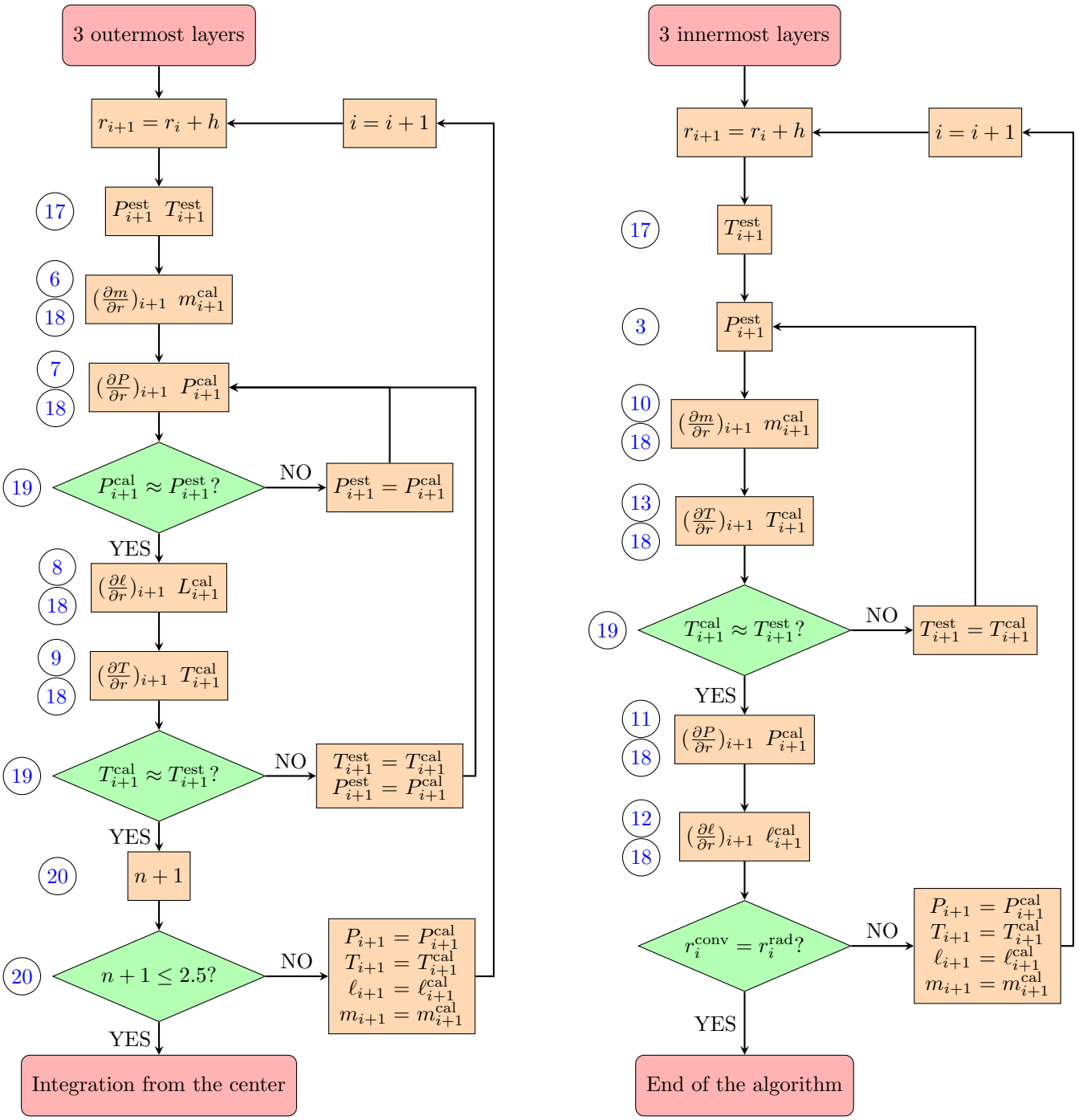

Figure 1: Flowchart of the implemented algorithm.

3 Model development

The numerical model of stellar interiors presented here was developed entirely in the `Python` programming language due to its simple syntax and versatility. As the project progressed, it became necessary to structure the code using object-oriented programming, which greatly facilitated the physical analysis by allowing results to be generated with simple commands and adapting the model to different stars. This significantly improved the model's accessibility, enabling its use without requiring an understanding of its internal structure. Ultimately, the project resulted in the creation of the *StellarPy* package, available on GitHub [5].

3.1 Introduction to the StellarPy package

The main goal of *StellarPy* is to provide open and easy access to the model, allowing anyone to use it without needing programming skills or an in-depth understanding of its internal workings. To achieve this, the package consists of four main components: the `Star` class, the optimization functions `error_table` and `find_minimum`, and the data set `hertzsprung-russell-data.csv`.

Moreover, as part of *StellarPy*'s commitment to accessibility, compatibility with `Quantity` objects from the Astropy package [6] has been included. These objects combine a numerical value with a physical unit, allowing users to input parameters in any unit system without worrying about manual conversions, as all unit handling is managed internally within the package.

Figure 2: *StellarPy* package logo.

3.2 The Star class

A `class` in `Python` is a structure that allows the definition of objects with specific characteristics and behaviors. It acts as a template from which instances are created, all sharing common properties and functionalities. In this context, the `Star` class is the core of *StellarPy*, as it enables the creation of representations of different stars, helping organize the code and improving its readability, reusability, and maintainability.

3.2.1 Attributes

To create an instance of `Star`, it is necessary to specify the variables that store information about the object. These represent its properties or state and distinguish it from other instances. Each `Star` instance includes the following attributes:

- `Mtot`: total mass of the star.
- `Rtot`: total radius of the star.
- `Ltot`: total luminosity of the star.
- `Tc`: central temperature of the star.
- `X`: mass fraction of hydrogen of the star.
- `Y`: mass fraction of helium of the star.

3.2.2 Methods

Each instance of `Star` has access to methods, which are functions defined within the class that allow performing actions or modifying the object's attributes.

`__init__(self, Mtot, Rtot, Ltot, Tc, X, Y)`

Initializes a new instance of `Star` with total mass `Mtot`, total radius `Rtot`, total luminosity `Ltot`, central temperature `Tc`, and mass fractions of hydrogen and helium `X` and `Y`, respectively. The input values can include physical units using `Quantity` objects. If no units are provided, the parameters are assumed to follow the model's internal unit system.

`__repr__(self)` and `__str__(self)`

These methods allow the `Star` instance to be represented as a text string, showing the values of its attributes using the model's internal units.

`get(self, variable='all', input_units=True, to_csv=False, name=None)`

Returns the requested data from the `Star` instance as a `Series` or `DataFrame` from the `pandas` package [7]. The `variable` argument accepts the string `'all'` to retrieve all computed variables, or one of `'r'`, `'P'`, `'T'`, `''l'`, `'m'`, `'rho'`, `'epsilon'`, or `'kappa'` to query a specific variable. The `input_units` argument can be set to `True` to display the data in the same units used when the instance was created, or `False` to use the model's internal units. Finally, if `to_csv` is `True`, a `.csv` file will be generated in the working directory with the name specified in `name`.

`parameters(self)`

This method provides a quick way to access the star's defining parameters by returning the attributes of the `Star` instance as a list in the format `[Mtot, Rtot, Ltot, Tc, X, Y]`.

`redefine(self, Mtot=None, Rtot=None, Ltot=None, Tc=None, X=None, Y=None)`

Allows redefining any of the attributes of the `Star` instance. As with the `__init__` method, it accepts inputs as `Quantity` objects, and parameters without units are interpreted using the model's internal units.

`visualize(self, x_axis='r', which=['P', 'T', 'l', 'm', 'rho'], merge=False, normalize=True, figsize=(8, 6))`

Generates graphical representations of the variables throughout the star. The `x_axis` and `which` arguments define the independent and dependent variables, respectively, and can be set to any combination of `'r'`, `'P'`, `'T'`, `'l'`, `'m'`, `'rho'`, `'epsilon'`, or `'kappa'`. The `merge` argument, when set to `True`, overlays all selected curves in a single figure. If `False`, each variable is plotted in a separate frame. Additionally, `normalize` can be set to `True` to normalize the data or `False` to keep the original scale. Finally, `figsize` allows customizing the size of the generated figure(s).

`error(self)`

Returns the total relative error (in percentage) computed from equation (21), based on the parameters of the `Star` instance used for the numerical calculation.

`TDD(self)`

Plots the star's trajectory on the Temperature-Density diagram. In this figure, Prialnik (2010, Chapter 7) [8] identifies four regions based on the dominant pressure source: ideal gas, electron degeneracy, relativistic electron degeneracy, and radiation pressure.

`HR(self)`

Plots the star on the Hertzsprung–Russell diagram using the effective temperature T_{eff}, which is estimated from the Stefan-Boltzmann constant $\sigma = 5.67040 \times 10^{-5}\,\text{erg}\,\text{cm}^{-2}\,\text{s}^{-1}\,\text{K}^{-4}$ and the relation:

$$L_{\text{total}} = 4\pi R_{\text{total}}^2 \sigma T_{\text{eff}}^4 \tag{28}$$

`__calculate(self)`

Computes the stellar interior numerical model as described in Section 2, returning both the calculated variables (as a `DataFrame`) and the total relative error of the computation. This allows other methods of the `Star` instance to access these values. The full code and internal workings are available in the *StellarPy* GitHub repository [5].

3.3 Optimization functions

These functions iteratively optimize stellar parameters to minimize the model's total relative error, converging toward the most physically consistent stellar representation.

`error_table(star, n, dR, dL, numbering=False)`

Given a `Star` instance, generates a $(2\texttt{n}+1)\times(2\texttt{n}+1)$ parameter grid where each cell represents a unique combination of total radius `Rtot` and luminosity `Ltot`. These parameters are varied around their initial values with step sizes `dR` and `dL`, which accept `Quantity` objects or default to model units. For each {`Rtot, Ltot`} pair, the method numerically solves for the central temperature `Tc` that locally minimizes the relative error, displaying these errors in the grid cells when `numbering` is set to `True`. The output returns the optimal parameter set {`Rtot, Ltot, Tc`} in model units and the corresponding minimized error.

`find_minimum(star, x0=None)`

Given a `Star` instance and initial parameters `x0` as `[Rtot, Ltot, Tc]`, returns and prints the parameter set {`Rtot, Ltot, Tc`} that minimizes the total relative error using model units, along with the error value. Internally, uses the `scipy` [9] `minimize` function for optimization. If no initial parameters are provided, the instance's attributes are used as defaults.

3.4 Data set for the Hertzsprung–Russell Diagram

For the proper representation of a model star in the Hertzsprung–Russell diagram using the `HR` method, a data set of stars populating the diagram is required. This data was obtained from Salman Chen's Kaggle repository [10] and the work of Pecaut & Mamajek (2013, ApJS, Vol. 208, id. 9) [11]. After cleaning and merging both data sets, *StellarPy* provides the file `hertzsprung-russell-data.csv`, which includes columns for effective temperature (`Teff`) and its logarithm (`log(Teff)`), the logarithm of total luminosity in solar units (`log(L/Lo)`), spectral class (`Spectral Class`), point size for plotting (`Plot Size`), and color representation (`Color`) for 351 distinct stars.

4 Model results

Having developed the tools to work with the model efficiently, it is possible to perform analyses and generate results with minimal code. To demonstrate this capability, this section presents a case study of a model star where every result and figure will be accompanied by the specific commands that produce it. The following initial attributes are assumed:

$$M_{\text{total}} = 5.0 \times 10^{33}\,\text{g} \qquad X = 0.80 \qquad Y = 0.16$$

$$R_{\text{total}} = 11.5 \times 10^{10}\,\text{cm} \qquad L_{\text{total}} = 40.0 \times 10^{33}\,\text{erg}\,\text{s}^{-1} \qquad T_{\text{c}} = 1.5 \times 10^{7}\,\text{K}$$

This way, initializing the selected star using the model's units through a `Star` instance is very easy.

```
star = Star(Mtot=5.0, Rtot=11.5, Ltot=40.0, Tc=1.5, X=0.80, Y=0.16)
```

4.1 Search for the minimum total relative error

Once the `Star` instance has been initialized and stored in a variable, it becomes necessary to evaluate the validity of the computed quantities throughout the star based on the specified attributes. For this purpose, the `error` method provides access to the total relative error of the computation.

```
print(star.error())
```

This yields an error of 103.03%, which is clearly improvable. One option to reduce this value involves successive iteration using the `error_table` function. To do this, the minimum found by the function is selected, the magnitudes of the variations `dR` and `dL` are reduced and the method is applied again. This is repeated until locating the minimum total relative error below a specified tolerance. An initial iteration of this procedure can be executed with just a single line of code, generating the image in Figure 3 and printing the results.

```
error_table(star=star, n=5, dR=0.5, dL=5.0, numbering=True)
```

```
Minimum found at (model units):

    Rtot = 11.5000     Ltot = 45.0000     Tc = 1.8649     Error:  4.6086 %
```

As observed, the parameters that minimize the error differ from the initial ones. Furthermore, while the error improves significantly, achieving a truly acceptable value would require numerous repetitions. In contrast, the `find_minimum` function converges faster and obtains better results in a single iteration. Thus, the parameters at the minimum can be used to update the optimal attributes via the `redefine` method.

```
Rmin, Lmin, Tmin, error_min = find_minimum(star=star)
star.redefine(Rtot=Rmin, Ltot=Lmin, Tc=Tmin)
```

```
Minimum found at (model units):
    Rtot = 11.2518     Ltot = 42.5688     Tc = 1.8569     Error:  0.0166 %
```

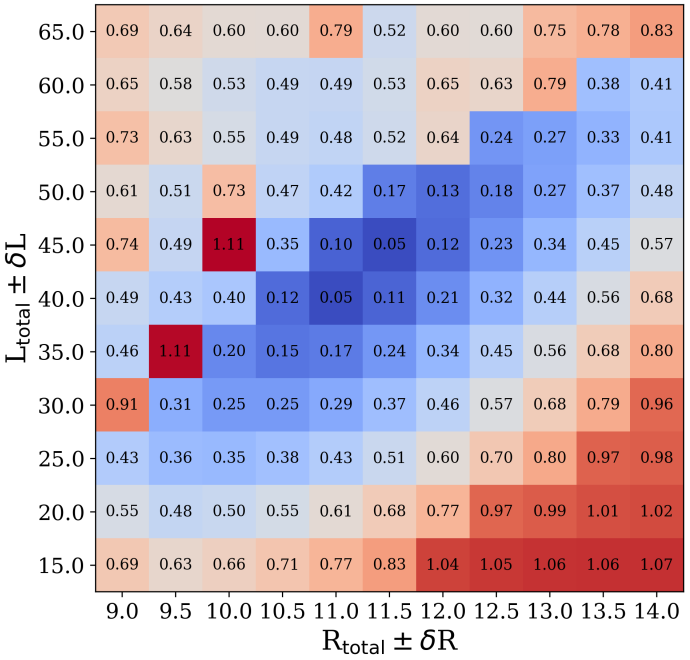

Figure 3: Summary table of total relative error values expressed on a scale from 0 to 1 for variations in total radius and luminosity.

4.2 Profiles of stellar variables

Once the error has been optimized in the `Star` instance, studying its properties becomes straightforward using the methods presented in Section 3.2.2. However, for each analysis shown here, it's crucial to consider the significant approximations introduced by accepting certain basic hypotheses to solve the stellar interior equations system.

4.2.1 Variation of P, T, ℓ, m and ρ with radius

The representation of P, T, ℓ, m y ρ versus r in Figure 4 allows for a comprehensive analysis of the stellar interior, validating the model and providing physical predictions about the studied star. To generate this image, only the following instruction is required:

```
star.visualize(merge=True, figsize=(10, 6))
```

- **Pressure**:

Decreases monotonically from the center to the surface, reaching its maximum value in the core, where it must be sufficient to counteract gravity and maintain hydrostatic equilibrium. At the transition between the convective and radiative zones, an inflection point and subtle change in slope are observed, which can be explained by convection's efficient energy transport mechanism and its association with a steeper pressure gradient.

- **Temperature**:

Decreases continuously from core to surface. Shows a steeper decline in the inner zone where the temperature gradient exceeds the adiabatic gradient, generating convective motions, and becomes smoother in the radiative zone. Therefore, the curve's shape indicates the numerical model correctly predicts the thermal distribution.

- **Luminosity**:

Increases with radius, starting from zero at the center and reaching its maximum at the surface, consistent with luminosity's definition as the total energy transported outward. The curve grows rapidly in the convective zone and stabilizes shortly after entering the radiative region. This behavior is expected due to the CN cycle's higher energy generation efficiency in inner layers compared to the proton-proton chain in the rest of the star.

- **Mass**:

Increases monotonically with radius until reaching the star's total mass. The curve shows a steeper slope in inner layers and gentler in outer ones, suggesting most stellar mass is concentrated in the interior region, accumulating half the mass within one-third of the radius.

- **Density**:

Decreases rapidly from core to surface, consistent with a star's expected structure. Furthermore, density becomes very low in the outermost layers, matching the concept of an extended, low-mass envelope.

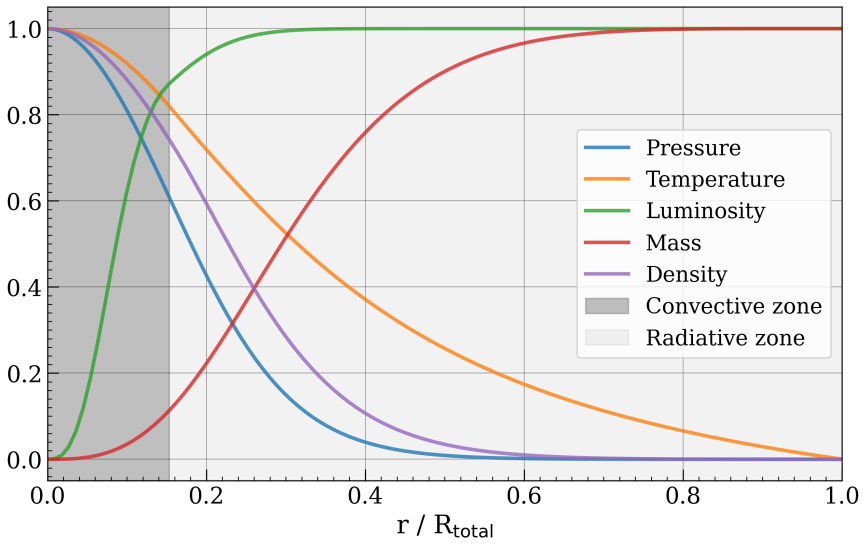

Figure 4: Variation of different physical quantities throughout the stellar radius.

4.2.2 Variation of opacity with radius

The analysis of opacity in stellar interiors is fundamental for understanding energy transfer and stellar structure. Although various mechanisms can contribute to κ, this study will focus exclusively on the conditions established by Kramers' law for bound-free transitions, as this is the only opacity source considered in the model construction. To generate the plot of its evolution versus radius, the `visualize` function is called.

```
star.visualize(which=['kappa'], normalize=True, figsize=(10, 6))
```

Figure 5 shows that in the star's innermost regions, opacity follows a stable trend, gradually increasing with radius. However, this increase becomes significant near the surface. This result is not unexpected, since the only considered opacity source comes from bound-free transitions. In the inner layers, high temperatures completely ionize the material, preventing these transitions from occurring. In contrast, in regions close to the atmosphere, partial ionization becomes more relevant, enhancing the processes responsible for bound-free opacity.

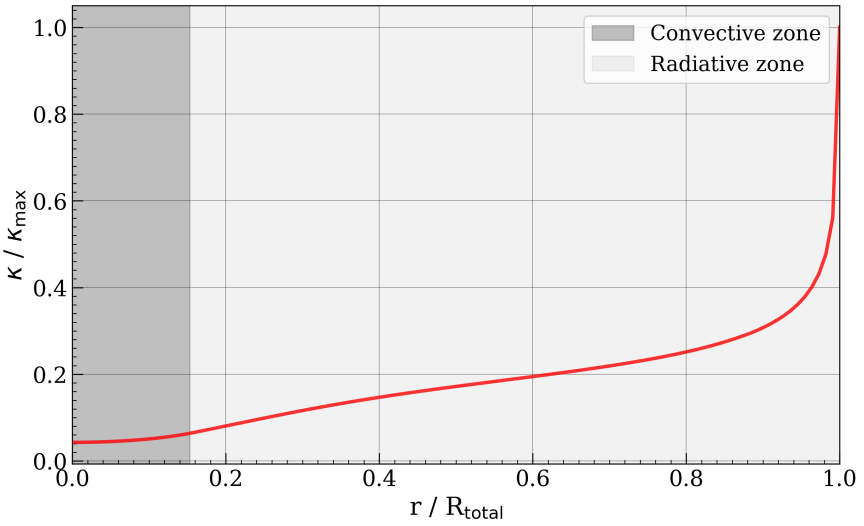

Figure 5: Variation of opacity throughout the stellar radius.

It's also interesting to evaluate the model's response to variations in stellar metallicity. For this purpose, a new `Star` instance can be created using the attributes of `star` via the `parameters` method, then increasing the hydrogen mass fraction by 0.005, thereby reducing metallicity from $Z = 0.040$ to $Z = 0.035$.

```
Mtot, Rtot, Ltot, Tc, X, Y = star.parameters()
new_star = Star(Mtot=Mtot, Rtot=Rtot, Ltot=Ltot, Tc=Tc, X=X+0.005, Y=Y)
Rmin, Lmin, Tmin, error_min = find_minimum(star=new_star)
```

```
Minimum found at (model units):

    Rtot = 11.0862     Ltot = 47.3190     Tc = 1.8765     Error:  0.0177 %
```

When comparing these results with those obtained for the `star` instance in Section 4.1, the effect is as expected. The influence of heavy element abundance on stellar properties comes primarily from opacity changes. For not very massive stars, decreasing metallicity reduces bound-free opacity, resulting in smaller convective zones and more transparent stars. Consequently, the stars become hotter, smaller, and slightly more luminous.

4.2.3 Variation of energy generation rate with radius

The study of the energy generation rate throughout a star allows to identify key properties such as energy transport through temperature gradients or to predict luminosity and surface temperature for later comparison with observations. Furthermore, the energy generation rate is directly linked to stellar nucleosynthesis and the chemical evolution of the universe. In this context, the `visualize` function enables plotting its profile against stellar radius.

```
11 star.visualize(which=['epsilon'], normalize=True, figsize=(10, 6))
```

The curve in Figure 6 shows a very pronounced maximum near the core, which is consistent with the extreme temperature and density conditions that favor intense nuclear reactions and where thermal gradients exceed the threshold for convective instability. In contrast, when the temperature gradient is no longer steep enough to drive convection, there is a rapid decrease in the energy generation rate outward, indicating that most energy is generated in the star's interior with very little coming from the outer layers.

On the other hand, it's important to highlight the use of the power law described in equation (5). This correctly reproduces the high temperature sensitivity of the energy generation rate, meaning small temperature differences generate large variations in ε. Additionally, since the exponent ν in the power law is significantly larger for the CN cycle, it becomes dominant in regions close to the core.

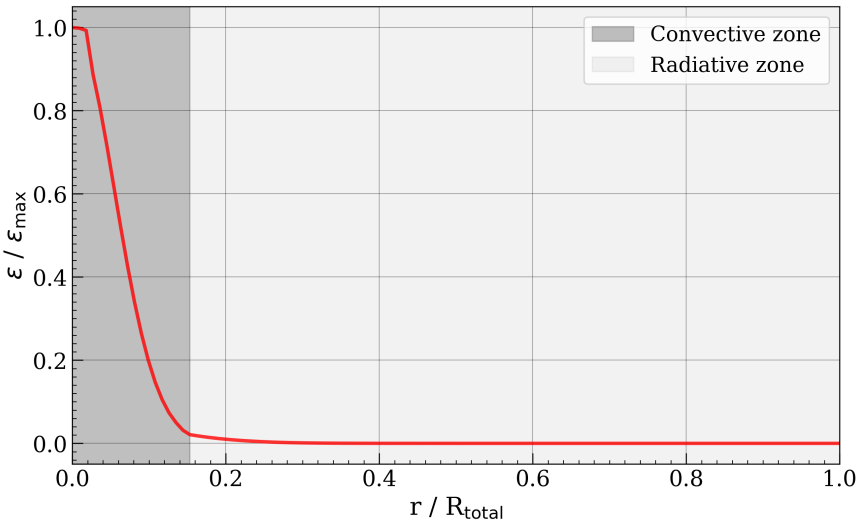

Figure 6: Variation of the energy generation rate throughout the stellar radius.

4.3 Temperature-Density Plane

The study of the temperature-density plane is particularly interesting due to the distinct zones identified according to the dominant pressure in the star: ideal gas pressure, degeneracy pressure, relativistic degeneracy pressure, and radiation pressure. A star will exhibit different behaviors and properties depending on which parts of its structure lie within each of these zones, thereby influencing its later evolution. Using the tools provided by *StellarPy*, creating this diagram is very simple.

```
12 star.TDD()
```

Figure 7 shows the expected result, since the nearly exclusive use of equation (2) to calculate pressure places the entire star in the ideal gas zone. On the other hand, it's interesting to note that while density varies by ten orders of magnitude between the center and surface, the material temperature shows much smaller variations.

Additionally, it's worth highlighting how the pressure supporting the star against gravity changes with depth. Thus, the innermost layers of the star, where density is highest, approach closer to the degeneracy zone. However, their separation from this region suggests the stellar core won't enter it during its evolution, which aligns with the fact that stars with masses similar to the one considered in this study don't experience the helium flash nor, consequently, electron degeneracy.

In contrast, the star's surface lies near the boundary where radiation pressure begins to become significant, indicating a non-negligible contribution from this component to hydrostatic equilibrium. This behavior is characteristic of massive stars, whose atmospheres experience mass loss due to this type of pressure, and reinforces the validity of the model used in this analysis.

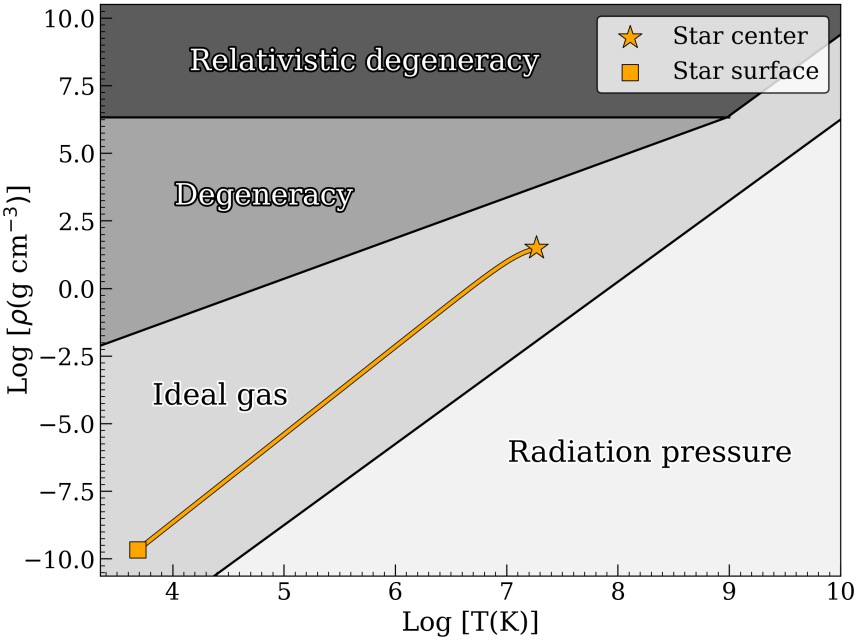

Figure 7: Trajectory of the model star in the Temperature-Density plane.

4.4 Hertzsprung–Russell Diagram

The Hertzsprung-Russell diagram is a fundamental tool in astronomy, as it allows stars to be classified according to their evolutionary stage, size, and surface temperature. Furthermore, it is essential for understanding the physical processes that determine stellar evolution, their internal composition, and their ultimate fate. With that in mind, the analysis is easily done with the model's implementation of equation (28). With just a single line of code, the simulated star can be plotted on the HR diagram, facilitating comparison with observational data and theoretical models.

```
13 star.HR()
```

For the generated `Star` instance, an effective temperature of $T_{\text{eff}} = 8288\,\text{K}$ is obtained, classifying the star as spectral type A, characterized by a bluish-white color. Moreover, Figure 8 shows how the model accurately predicts the star's position on the main sequence as a massive object in the early evolutionary stages with slow hydrogen fusion. Similarly, the star's placement in the diagram allows us to predict that, since the core doesn't enter degeneracy and there's no helium flash, the star will leave the main sequence smoothly, transitioning to the subgiant phase. Finally, it's important to note that the model's scope is limited to less-evolved main sequence stars, particularly those with masses greater than $1.3\,M_{\odot}$, since according to Kippenhahn et al. (2012, Chapter 22) [1], this mass range corresponds to stars with convective cores and radiative envelopes.

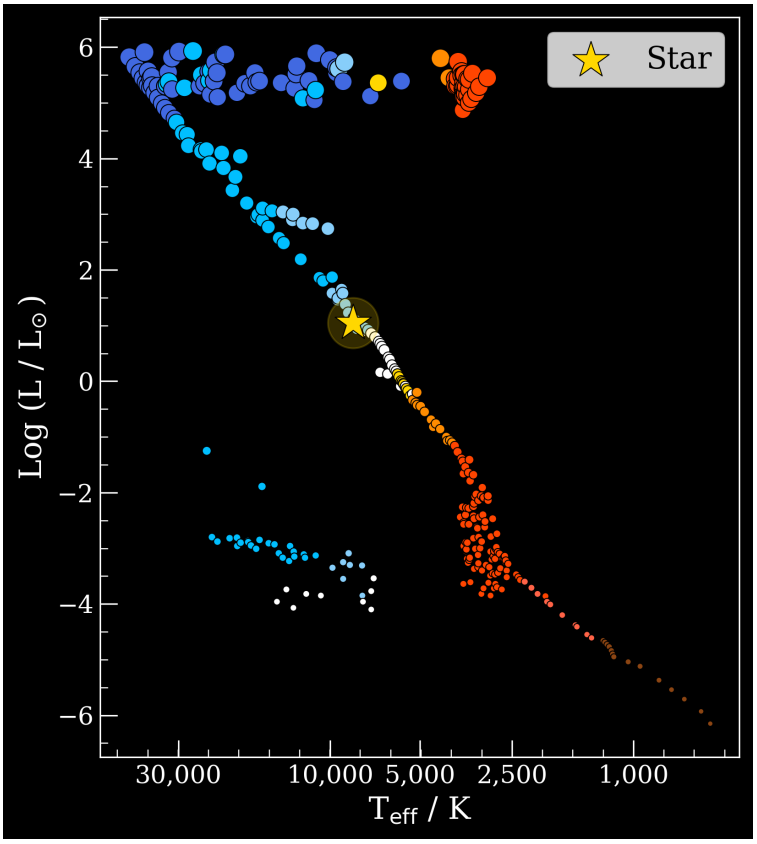

Figure 8: Position of the model star on the Hertzsprung–Russell diagram.

5 Model limitations

Throughout Section 4, the performance of the developed stellar interior model has been demonstrated in generating significant results and facilitating analysis of different star types. However, understanding the scope of its predictions and their validity is crucial for proper interpretation of these results. To study its limitations, it can be useful to attempt to model the star Altair and compare the obtained solution with the model developed in Bouchaud et al. (2020) [12]. All initial parameters will be taken from the article except for the central temperature, which will be assumed the same value as the one from the star modeled previously. Additionally, since Altair is a rapidly rotating star, the total radius will be calculated as the average between the equatorial radius $R_{\text{equatorial}} = 2.007\,R_\odot$ and polar radius $R_{\text{polar}} = 1.565\,R_\odot$.

$$M_{\text{total}} = 1.86\,M_\odot \qquad X = 0.710 \qquad Y = 0.271$$

$$R_{\text{total}} = 1.786\,R_\odot \qquad L_{\text{total}} = 10.6\,L_\odot \qquad T_{\text{c}} = 1.5 \times 10^7\,\text{K}$$

In this case, using `Quantity` objects along with the `find_minimum` function greatly simplifies accessing the desired result.

```
14 altair = Star(Mtot=1.86*u.solMass, Rtot=1.786*u.solRad, Ltot=10.6*u.solLum, Tc=1.5,
15                X=0.710, Y=0.271)
16 Rmin, Lmin, Tmin, error_min = find_minimum(star=altair)
```

```
Minimum found at (model units):

     Rtot = 8.6904     Ltot = 33.2773     Tc = 1.8971     Error:  0.0355 %
```

From these results, only the total radius and luminosity can be compared, since Altair's central temperature is unknown and both mass and chemical composition are fixed parameters. Therefore, when expressed in solar units, the values are:

$$R_{\text{total}} = 1.249\,R_\odot \qquad L_{\text{total}} = 8.7\,L_\odot$$

Comparing with the previously presented parameters, while the model achieves correct orders of magnitude, its predictions for Altair's radius and luminosity differ significantly from expected values, producing a smaller and less luminous star.

As mentioned several times, these discrepancies primarily arise from the simplifying assumptions adopted in Section 1. Specifically, Altair's rapid rotation prevents our model from accurately reproducing its properties.

However, this is not the only constraint. Some of the most significant are: the model assumes spherical symmetry, excluding rotation and magnetic field effects, leading to inadequate descriptions of rapidly rotating stars. It presumes local thermodynamic equilibrium, a condition not always met in outer layers or stellar atmospheres where radiative and convective processes can be more complex. It employs a polytropic treatment that oversimplifies convective behavior in inner regions, limiting accuracy for stars with significant convective zones. It uses Kramer's laws for opacity modeling, which doesn't precisely represent complex atomic and molecular interactions. Finally, it approximates the energy generation rate through a power law, introducing errors in predicting stellar structure and evolution.

6 Conclusions

Throughout this project, the main objective have been successfully achieved: developing a numerical model to describe a star's internal structure from physical parameters. This accomplishment involved both theoretical work (studying and understanding the fundamental equations of stellar interiors and applying the necessary simplifying assumptions for their solution) and practical implementation (through a functional numerical algorithm and development of the corresponding code).

The project evolved from a personal `Python` script into an accessible user-friendly software. Furthermore, illustrative figures and detailed results analysis were produced to evaluate the model's behavior across stellar configurations. Additionally, model's limitations and conditions under which its application is not suitable were established, including cases like massive stars, advanced evolutionary stages, radiative-core stars, and binary systems.

As a solid first step in stellar modeling, this project provides useful results within its scope while demonstrating the need for more sophisticated models. It can be used as a convenient tool in scientific research for obtaining estimates that are within the right order of magnitude. In conclusion, this software establishes a foundation for future simulations to better describe stellar structure and evolution across a wider range of astrophysical contexts.

References

[1] Kippenhahn R. et al., "Stellar Structure and Evolution", Astronomy & Astrophysics Library, Springer-Verlag, 2012, Berlin, Heidelberg.

[2] Schwarzschild M., "Structure and Evolution of the Stars", Dover Publications, 1965, New York.

[3] Novotny E., "Introduction to Stellar Atmospheres and Interiors", Oxford University Press, 1973, New York.

[4] Cardiel N. & Pascual S., "Elaboración de un modelo numérico de interior estelar", 2024, Madrid.

[5] Cerviño N., "StellarPy", GitHub Repository: https://github.com/gallati/stellarpy

[6] Astropy Collaboration. Available in: https://www.astropy.org

[7] Pandas: Python Data Analysis Library. Available in: https://pandas.pydata.org

[8] Prialnik D., "An introduction to the theory of stellar structure and evolution", 2010, Cambridge University Press.

[9] SciPy: Scientific Computing Tools for Python. Available in: https://scipy.org

[10] Chen S., "Hertzsprung-Russell diagram", Kaggle. Retrieved from https://www.kaggle.com/code/salmanhiro/hertzsprung-russell-diagram/input

[11] Pecaut M. & Mamajek J, "Intrinsic Colors, Temperatures, and Bolometric Corrections of Pre-main-sequence Stars", The Astrophysical Journal Supplement, Volume 208, Issue 1, article id. 9, 22 pp. Retrived from https://www.pas.rochester.edu/~emamajek/EEM_dwarf_UBVIJHK_colors_Teff.txt

[12] Bouchaud K. et al. "A realistic two-dimensional model of Altair", 2020, Astronomy and Astrophysics. 633: A78. arXiv:1912.03138.